IN THE CLAIMS

1. Please cancel Claims 5-10 and 15-16 without prejudice, amend Claims 1, 3, 11, 12, 17, 18, and 19, and add new Claims 20-42 as follows:

1. (Currently amended) A method of formatting a plurality of strings of data derived from respective ones of a plurality of data processors, comprising:

initializing the data in each of said strings;

finding the differences between said strings; and

providing said differences in a display to a user;

wherein said act of finding the differences comprises:

identifying groups of said data within said strings that are identical across said plurality of input strings; and

identifying groups of said data within said strings that appear in the same order within all of said strings.

2. (Previously presented) The method of Claim 1, wherein the act of initializing comprises:

creating a symbol table having a plurality of symbol numbers associated therewith;

creating said symbol array having at least one element for each of said strings;

for each of said input strings, determining whether said each string is present in said symbol table; and

if said each string is present in said symbol table, obtaining at least one symbol number for said string from said symbol table.

3. (Currently amended) A method of analyzing a plurality of strings of data derived from at least one data processing device, comprising:

initializing said data, said act of initializing including creating a symbol array;

analyzing said strings of data based at least in part on said symbol array; and

identifying at least one relationship between one or more of said data within one or more of said strings;

wherein said act of identifying comprises:

identifying groups of said data within said strings that are identical across said plurality of input strings; and

identifying groups of said data within said strings that appear in the same order within all of said strings.

4. (Previously presented)   The method of Claim 3, wherein the act of initializing comprises:

creating a symbol table having a plurality of symbol numbers associated therewith;

creating said symbol array having at least one element for each of said strings;

for each of said input strings, determining whether said each string is present in said symbol table; and

if said each string is present in said symbol table, obtaining at least one symbol number for said string from said symbol table.

5.-10. (Cancelled)

11. (Currently amended)   A storage device, comprising:

a computer readable medium;

a computer program stored on said computer readable medium, said program being adapted for analyzing a plurality of strings of data derived from at least one data processing device according to the method comprising:

initializing said data, said act of initializing including creating a symbol array;

analyzing said strings of data based at least in part on said symbol array; and

identifying at least one relationship between one or more of said data within one or more of said strings;

wherein said act of identifying comprises:

identifying groups of said data within said strings that are identical across said plurality of input strings; and

identifying groups of said data within said strings that appear in the same order within all of said strings.

12. (Currently amended)   A data processing device, comprising:

a processor adapted to process digital data and execute a computer program;

a storage device in data communication with said processor, said storage device comprising:

a computer readable medium; and

a computer program stored on said computer readable medium, said program being adapted for analyzing a plurality of strings of data derived from at least one data processing device according to the method comprising:

initializing said data, said act of initializing including creating a symbol array;

analyzing said strings of data based at least in part on said symbol array; and

identifying at least one relationship between one or more of said data within one or more of said strings;

wherein said act of identifying comprises:

identifying groups of said data within said strings that are identical across said plurality of input strings; and

identifying groups of said data within said strings that appear in the same order within all of said strings.

13. (Previously presented)   The device of Claim 12, further comprising a plurality of data interfaces adapted to receive said data strings from respective ones of a plurality of software processes running on respective ones of a plurality of data processors.

14. (Previously presented)   The device of Claim 13, further comprising said plurality of data processors.

15.-16. (Cancelled)

17. (Currently amended)   A method of evaluating the operation of a plurality of software processes running on respective ones of a plurality of digital processors, comprising:

generating a first data string using a first of said plurality of software processes;

generating a second data string using a second of said plurality of software processes;

generating a third data string using a third of said plurality of software processes;

inputting said first and second and third data strings into a debug software process;

analyzing said first and second <u>and third</u> data strings using said debug process; and

evaluating the operation of said processes based at least in part on said act of analyzing.

18. (Currently amended)   The method of Claim 17, wherein the act of analyzing comprises:

(i)   identifying common patterns within at least one first portion of said first and second <u>and third</u> data strings; and

(ii)   identifying differences within at least one second portion of said first and second <u>and third</u> data strings.

19. (Currently amended)   A multi-processor integrated circuit device, comprising:

a first processor core adapted to run a first software process, said first software process adapted to generate a first string of data;

a second processor core adapted to run a second software process said second software process adapted to generate a second string of data;

at least one data interface, wherein said first and second processors respectively transfer data comprising said first and second strings to an external debug process adapted to identify similarities and differences between the operation of said first and second ~~softare~~ <u>software</u> processes on said first and second processors, via said at least one interface;

<u>wherein said debug process identifies the similarities and differences by:</u>

<u>identifying groups of said data within said first and second strings that are identical across at least both of said strings; and</u>

<u>identifying groups of said data within said first and second strings that appear in the same order within at least both of said strings.</u>

20. (New)   The method of Claim 1, wherein said strings each comprise a plurality of lines, and said method further comprises:

forming a plurality of groups of lines, wherein a group comprises a sequence of lines that are the same in all of said strings; and

recursively analyzing, in order:

a first region of all the strings that appears before the first of said plurality groups;

each of a plurality of second regions occurring between two of said plurality of groups; and

a third region following the last of said plurality of groups.

21. (New)   The storage device of Claim 11, wherein said strings each comprise a plurality of lines, and said method further comprises:

forming a plurality of groups of lines, wherein a group comprises a sequence of lines that are the same in all of said strings; and

recursively analyzing, in order: a first region of all the strings that appears before the first of said plurality groups;

each of a plurality of second regions occurring between two of said plurality of groups; and

a third region following the last of said plurality of groups.

22. (New)   The integrated circuit device of Claim 19, wherein said strings each comprise a plurality of lines, and said debug process is further adapted to:

form a plurality of groups of lines, wherein a group comprises a sequence of lines that are the same in all of said strings; and

recursively evaluate, in order:

a first region of all the strings that appears before the first of said plurality groups;

each of a plurality of second regions occurring between two of said plurality of groups; and

a third region following the last of said plurality of groups.

23. (New)   Computer storage apparatus having a media adapted to store a plurality of data, said data comprising a computer program, said computer program being adapted to process a plurality of input streams to identify groups of elements within said streams that are:

(i)   identical across said plurality of input streams; and

(ii)   which appear in the same order within all of said streams.

24. (New)   The apparatus of Claim 23, wherein said input streams comprise a plurality of lines, and said program is further adapted such that if no groups of elements are identified, said program:

(i)   compares all of said streams and lines in order of occurrence; and

(ii)   provides a representation of each unique line prefixed by those streams that contain said unique line.

25. (New)   The apparatus of Claim 23, wherein said input streams comprise a plurality of lines, and said program is further adapted such that if one or more groups are identified, said program recursively analyzes a first region in all the streams before a first of said one or more groups.

26. (New)   The apparatus of Claim 25, wherein when said program identifies two or more groups, said program recursively analyzes the region between each of individual sequential ones of said groups in order, and the region following a last of said two or more groups.

27. (New)   A method of processing a plurality of input streams of data elements, said method comprising:

identifying groups of said elements within said streams that are identical across said plurality of input streams; and

identifying groups of said elements within said streams that appear in the same order within all of said streams.

28. (New)   The method of Claim 27, wherein said elements of said input streams comprise a plurality of lines, and said method further comprises, where no groups of elements are identified:

comparing all of said streams and lines in order of occurrence; and

providing a representation of each unique line, said representation being prefixed by those streams that contain said unique line.

29. (New)   The method of Claim 27, wherein said input streams comprise a plurality of lines, and said method comprises recursively analyzes a first region in all the streams before a first of said one or more groups.

30. (New)   The method of Claim 29, wherein when two or more groups are identified, recursively analyzing the region between each of individual ones of said groups in order, and also the region following a last of said two or more groups.

31. (New)   The method of Claim 27, wherein said processing of a plurality of input streams comprises processing two or more streams derived from respective ones of digital processors.

32. (New)   The method of Claim 30, wherein said processing of a plurality of input streams comprises processing two or more streams derived from respective ones of digital processors.

33. (New)   A computerized method for analyzing a plurality of input streams having a plurality of elements arranged in lines, comprising:

identifying any of said streams that have no lines in common with any other stream; and

removing such streams from further analysis.

34. (New)   The method of Claim 33, further comprising:

forming one or more groups of lines, where a group comprises a sequence of lines that are the same in all of said input streams.

35. (New)   The method of Claim 34, wherein said act of forming comprises at least discovering unique lines that occur exactly once and in every one of said input streams.

36. (New)   The method of Claim 35, wherein said act of forming further comprises expanding at least one line surrounding each of said unique lines as long as said at least one line contains elements that match in all of said input streams.

37. (New)   The method of Claim 34, further comprising verifying that all of said groups occur in the same order in all of said streams.

38. (New)   The method of Claim 33, further comprising evaluating all of said lines in all of said streams to determine the number of times each element occurs in each stream.

39. (New)   The method of Claim 38, further comprising:

determining which of said plurality of streams have less than a predetermined portion of its elements appear in no other stream; and

discarding all streams other than those identified during said act of determining.

40. (New)   The method of Claim 39, further comprising sequentially incrementing through each of the lines of all non-discarded streams, said act of incrementing further comprising:

identifying each distinct element occurring at the selected line number; and

identifying those streams containing that element.

41. (New)   A computerized method for analyzing a plurality of input streams having a plurality of elements arranged in lines, said streams being derived from respective ones of digital processors, the method comprising:

identifying any of said streams that have no lines in common with any other stream;

removing such identified streams from further analysis;

using the remaining streams, forming a plurality of groups of lines, wherein a group comprises a sequence of lines that are the same in all of said remaining input streams; and

recursively analyzing, in order:

a first region of all the streams that appears before the first of said plurality groups;

each of a plurality of second regions occurring between two of said plurality of groups; and

a third region following the last of said plurality of groups.

42. (New)   The method of Claim 33, wherein said act of recursively analyzing comprises producing additional ones of said groups in said first, second or third regions.